周期性分布式仿真系统中的同步问题研究

许建峰, 朱晴波, 胡宁宁, 谢 立
(南京大学计算机科学与技术系, 江苏南京 210093)

摘 要: 本文分析了分布式仿真模型SDAG 中的同步机制, 得到了事件同步的充分条件, 提出了MATW 和 GDSM 算法, 通过对各个实体上仿真任务运行时间以及输入消息到达时差的预测, 动态调整同步延迟. 测试结果表明新算法达到了改善系统同步性能的目的.

关键词: 仿真; 分布式; 同步
中图分类号: TP316.4 文献标识码: A 文章编号: 0372-2112 (2002) 05-0680-05

Study of Synchronization in Periodically Distributed Simulation System

XU Jiarr-feng, ZHU Qing-bo, HU Ning-ning, XIE Li
(*Department of Computer Science and Technology, Nanjing University, Nanjing, jiangsu 210093, China*)

Abstract: This paper analyzes the synchronization mechanism of distributed simulation model SDAG and obtains the sufficient conditions for events synchronization. Then MATW and GDSM algorithms have been proposed which can adjust the synchronization delay by predicting the runtime difference of simulation and the arriving time difference of input messages of all simulation entities. The test result has showed the good performance of the new algorithms in improving system synchronization.

Key words: simulation; distributed systems; synchronization

1 引言

针对分布式仿真系统中的同步问题, 人们提出了众多解决方法[1], 概括而言, 可分为保守(Conservative) 算法[2] 与乐观(Optimistic) 算法[3] 两大类. NM(Null Message) [4] 是经典的保守协议, 它采用空消息机制有效地避免死锁的产生, 因而得到了广泛应用. TW(Time Warp) [5] 是最著名的乐观协议, 已经在很多系统中实现, 现在仍然受到人们的极大关注. 无论保守还是乐观方法都有其局限之处, 为此人们一直致力于寻求各种折衷, ATW(Adaptive TW, ATW) 是近年来人们提出的一类富有成效的同步算法[6], 但此类方法只能通过事件延迟来避免其所在节点上出现的因果律违背, 因而适应性受到很大限制, 本文所讨论的一类特殊有向无环图(Special Directed Acyclic Graph, SDAG) 结构的仿真系统即难以直接运用 ATW 方法. 基于上述原因, 我们提出了修正的 ATW(Modified ATW, MATW) 和全局动态同步算法(Global Dynamic Synchronization Method, GDSM), 通过对各个实体上仿真任务运行时间以及输入消息到达时差的预测, 动态调整同步延迟, 达到改善周期性分布式仿真系统同步性能的目的.

2 SDAG 结构

2.1 RDSP 系统及其典型应用实例

在开发的实时分布式仿真平台 RDSP(Real-time Distributed Simulation Platform) 中, 仿真应用软件的运行单位称为实体, 分配在分布式环境中的各台处理节点上, 各个实体之间的数据流构成一有向无环图, 且按照一定的周期重复运行, 共同完成系统仿真任务. 在仿真过程中, 实体的运行时间及实体之间的通信时间均为平稳变化的随机参量[7]. 图 1 便是一个在 RDSP 中运行的机载脉冲多普勒雷达主杂波跟踪系统(FCT) 的简化模型.

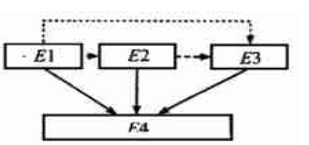

图 1 同步实例

2.2 数据匹配问题

文[1] 提出的 NM 算法基于一个前提, 即系统中存在唯一的无输入实体(source, 称为源) 和唯一的无输出实体(sink, 称为末端), 在 RDSP 中, 无论在何种情形也总能通过空消息的加入来满足这个条件.

定义(周期) 仿真实体从源开始, 按数据流程依次运行至末端为止的过程称为周期.

定义(匹配性) 一个实体在一个周期中处理的所有输入消息均产生于同一周期, 则称这些输入消息之间是匹配的; 反之, 则称这些输入消息之间是失配的.

在 FCT 系统中, 当系统中出现消息失配时, 仿真结果的精度将会大幅下降. 因此在仿真过程中必须保证消息之间的匹配性, 但这往往由于实体运行及通讯延迟的随机特性而受

收稿日期: 2001-04-09; 修回日期: 2001-10-06
基金项目: 国家 863 项目(No. 863-306-ZT02-03-01)

到影响, 具体原因有:

(1) 滞后效应

当实体运行采用时间驱动方式时, 某些输入消息会因处理及通信上的滞后而未能及时到达, 这样便会导致消息的失配.

(2) 超前效应

当实体运行采用事件驱动方式时, 一个实体必须在所有输入信箱非空时才能运行, 这样便会出现某些消息的超前到达而导致消息失配.

2.3 SDAG 结构的主要特征

根据上面的讨论, 可以抽象出 SDAG 结构分布式仿真系统的主要特征:

(1) 系统由分布在各个计算节点上的仿真实体组成, 所有仿真实体均按周期往复运行, 并且实体间的数据流程构成有向无环图;

(2) 系统中存在唯一的源和末端;

(3) 实体的运行时间及实体之间的通讯时间均为平稳变化的随机参量;

(4) 实体之间的通信通过信箱实现, 同一信箱中新邮件覆盖旧邮件, 因而在仿真过程中存在消息失配的可能;

(5) 对系统中可能引起消息失配的的实体采用事件驱动机制, 可避免滞后效应导致的消息失配, 但需要通过实体之间的同步算法来解决超前效应导致的消息失配问题.

3 同步问题及典型算法

3.1 GVT

全局虚拟时间(Global Virtual Time, GVT) 是描述和实现分布式仿真系统同步的基础, 在 RDSP 中, 任一时刻每一个实体以及实体之间通信的消息都有唯一的 GVT 标识, 具体定义如下:

定义(**GVT**)　仿真开始时所有实体的 GVT 均为 0; 当源实体完成了第 i 周期的仿真而未完成第 $i+1$ 周期的仿真时, 其 GVT 为 i; 设 m_{ij} 为实体 e_i 发往实体 e_j 的消息, 则

$$GVT(m_{ij}) = GVT(e_i);$$

设实体 e_j 已完成对一组输入消息 $m_{i1j}, m_{i2j}, \cdots, m_{inj}$ 的仿真处理, 而未完成其后的仿真处理, 于是

$$GVT(e_j) = \max(GVT(m_{i1j}), GVT(m_{i2j}), \cdots GVT(m_{inj})).$$

在系统仿真过程中, 要求实体之间的消息均含有自身的 GVT, 因此上述定义是可实现的.

3.2 消息失配与同步问题

因为 SDAG 中采用事件驱动机制, 并且每一实体启动运行的条件是其所有输入信箱非空, 所以通过数学归纳法我们能够方便地得到:

命题 **1**　对 SDAG 中任一异于源的实体 e_j, 若系统中未出现消息失配, 则当 e_j 的所有输入消息 $m_{i1j}, m_{i2j}, \cdots, m_{inj}$ 均已到达且未被处理时,

$$GVT(m_{i1j}) + 1 = GVT(m_{i1j}) = GVT(m_{i1j}) = \cdots = GVT(m_{i1j}).$$

上式通过 GVT 表达式显示了 SDAG 中事件之间的因果关系. 由于不采用时间驱动机制, 因而系统中不会出现滞后效应导致的消息失配, 但却会由于某些消息的"超前"到达而引起消息失配, 故也易于得到如下结论:

命题 **2**　SDAG 中出现消息失配的充分必要条件是存在实体 e_j 及其未被处理的输入消息 m_{ij}, 使

$$GVT(m_{ij}) \leqslant GVT(e_j).$$

可见, SDAG 中的消息失配问题等价于系统的同步问题, 并且一个实体上的事件只能引起其它实体上出现因果律的违背.

3.3 NM 同步算法

保守同步方法的经典之一是 NM 算法, 在 SDAG 中也能够采用 NM 算法. 假设消息 m_{ij} 的输出实体为 e_i, 输入实体为 e_j, 则称 e_i 为 e_j 的前驱实体, e_j 为 e_i 的后继实体. 于是, NM 算法的基本思想便是要求每个后继实体在完成一次仿真计算后向其所有前驱实体发送一空消息, 每一实体只有在收到其所有后继实体的空消息后方能发送下一组仿真消息. 显然这样的同步机制能够解决消息失配问题, 且由于 SDAG 中的仿真消息流程构成有向无环图, 因而不会在系统中形成"死锁". 虽然 NM 算法解决了 SDAG 中的同步问题, 但由于其基本思想过于"保守", 因而仿真过程中系统的并行度难以得到提高. 以 FCT 系统为例, 若假设 E_1、E_2、E_3 和 E_4 在一个周期中的仿真运行时间均为一个时间单位, 它们分布在不同的仿真节点上, 并且实体之间的通信延迟可以忽略不计, 于是可以得到表 1 所示的 GVT 分布, 其中 t 所在行为时间单位计数, 以下各行则为各个时段中各个实体的 GVT 标识. 可见, 尽管通信延迟已是分布式系统中最理想的情况, 系统仍需要 4 个时间单位才能完成一个周期的仿真, 其并行度只等同于一个串行系统.

表 **1**　NM 算法下的 GVT 分布

t	0	1	2	3	4	5	6	…
E_1	0	1	1	1	1	2	2	…
E_2	0	0	1	1	1	1	2	…
E_3	0	0	0	1	1	1	1	…
E_4	0	0	0	0	1	1	1	…

3.4 TW 同步算法

TW 同步算法是乐观方法的典型范例, 在 SDAG 中采用 TW 算法不需要增加实体之间的空消息, 但必须加入 GVT 检测机制, 即除源外的每一实体在每一周期仿真开始前需要对自身 GVT 和所有输入消息 GVT 进行比较, 若输入消息 GVT 之最小值小于或等于自身 GVT, 则应发出"反消息", 通知有关实体恢复至 GVT 为该最小值减 1 时的状态再开始仿真,"反消息"的发送路径以及需要进行回滚处理的实体范围与具体的回滚算法有关, 由于篇幅限制, 在此不再赘述. 若 FCT 系统中的所有条件均与 3.3 节所述相同, 忽略各个实体中保存现场的处理开销后采用 TW 算法可以得到表 2 所示的 GVT 分布, 可见其在系统同步未被破坏时能使并行度达到最高. 但如果同步遭到破坏, 回滚处理的代价将取决于系统结构和规模、算法复杂性、通信延迟等多种因素, 完全有可能使高度并行仿真产生的效益丧失殆尽.

表 2 TW 算法下的 GVT 分布

t	0	1	2	3	4	5	6	…
E_1	0	1	2	3	4	5	6	…
E_2	0	0	1	2	3	4	5	…
E_3	0	0	0	1	2	3	4	…
E_4	0	0	0	0	1	2	3	…

4 MATW 与 GDSM 同步算法

由于 SDAG 中实体运行及通信时间均为平稳变化的随机参量,即一方面呈现出很大的随机性,另一方面其短暂时间内的形态参数也表现出一定的相关性,因而能够通过相关参数的预测来改善系统的关键性能指标, ATW 正是一类根据这样的原理在 NM 与 TW 之间进行自适应调整的同步算法,对于一些满足特定条件的系统已经取得较为理想的结果. 然而, ATW 算法仅通过节点对自身事件的延迟来消除出现在本节点上的因果律违背,所以不能直接应用于 SDAG,故必须进一步地对系统的结构与时间特征进行分析方能获得更好的解决方案.

4.1 同步等待延迟

定义(源始路径) 对于 SDAG 中除源(E_0)以外的任一实体 E_i,若存在仿真消息从 E_0 开始,经 $E_{i1}, E_{i2}, \cdots, E_{in}$ 到达 E_i,则我们称实体组($E_0, E_{i1}, \cdots E_{in}, E_i$)为 E_i 的一条源始路径,简记为 LE_i. 因为一个实体可能存在多条源始路径,用 LE_{ij} 表示 E_i 的第 j 条源始路径.

定义(源始路径延迟) 对于源始路径 $LE_{ij}=(E_0, E_{i1}, \cdots E_{in}, E_i)$,将从 E_0 开始仿真至 E_i 收到 E_{in} 仿真消息的时间延迟称为实体 E_i 的源始路径延迟,记为 $D(LE_{ij})$.

在分布式仿真系统中,因为难以在所有节点之间建立统一的绝对时钟,所以无法要求各个实体都能获得精确的源始路径延迟,但是通过各个实体所在节点的内部时钟,易于得到任一实体各条源始路径延迟的相对时差. 特别地,将 E_i 的最大源始路径延迟与 $D(LE_{ij})$ 之间的差值记为 $d(LE_{ij})$. 同时,对于 SDAG 中任意一个实体 E_i,也易于得到其一个周期中的仿真运行时间 $R(E_i)$. 需指出的是,这里的各类延迟参数都是随仿真周期变化的,因此均以下标 n 表示第 n 周期中的延迟参数. 这样,对于 *TW* 算法我们易于证得:

命题 3 设 E_k 是 E_i 的前驱实体,且属于 E_i 的源始路径 LE_{ij},则当

$$R_{n+1}(E_k) > R_n(E_i) + d_{n+1}(LE_{ij})$$

时, E_k 发送的第 $n+1$ 周期仿真消息不会导致 E_i 的输入消息失配.

因此采用 TW 算法的 SDAG 中不出现因果律违背的充分条件是上述条件在任一周期对所有实体均成立. 由于系统中还存在其它延迟,因而这个条件并不是必要条件. 另一方面,正由于条件的苛刻,难以为大多数系统所满足,也说明了 TW 算法的局限性. 为此,需要引入同步等待延迟 $S_n(M_{ki})$,它是前驱实体 E_k 在完成了第 n 周期仿真任务后为实现同步而暂缓向后继实体 E_i 发送输出结果的等待时间,进而能够得到一个关于同步等待延迟的更为实用的结果:

命题 4 设 E_k 是 E_i 的前驱实体,且属于 E_i 的源始路径 Le_{ij},则当同步等待延迟

$$S_{n+1}(M_{ki}) > R_n(E_i) + d_{n+1}(LE_{ij}) - R_{n+1}(E_k)$$

时, E_k 发送的第 $n+1$ 周期仿真消息不会导致 E_i 的输入消息失配.

4.2 MATW 算法

命题 4 表明,同步问题可以转化为各个实体上仿真运行时间和源始路径延迟的确定问题,其中, $R_{n+1}(E_k)$ 延迟发生在 $S_{n+1}(M_{ki})$ 之前,因而可以直接获得. 每个周期的 $d(LE_{ij})$ 与 $R(E_i)$ 都能够在事件发生后由 E_i 直接得到,当若干周期前的这些数据被回送至 E_k 后,根据延迟参数的相关特性,运用预测算法 E_k 能够获得 $d_{n+1}(LE_{ij})$ 与 $R_n(E_i)$ 的估计值,便可最后确定 $S_{n+1}(M_{ki})$ 的下限. 至于具体的预测算法,均值估计、指数平滑、Kalman 滤波[6]以及 α-β-γ 滤波[7]等都是理论成熟且实际应用中取得很好效果的预测估计算法,可以根据实际系统的具体模型来选择,本文不再对此进行进一步的讨论. 根据上述结论并结合 ATW 的基本思想,提出 MATW,即修正的 ATW 算法如下:

①系统仿真启动时,采用 NM 算法,后继实体回送前驱实体的空消息中加入本周期的 $d(LE_{ij})$ 与 $R(E_i)$ 数据;

②前驱实体根据已经接收的后继实体 $d(LE_{ij})$ 与 $R(E_i)$ 时间序列分别对后续 $d(LE_{ij})$ 与 $R(E_i)$ 进行预测估计;

③在预测估计算法的初始化过程结束后,前驱实体根据命题 4 计算 $S(M_{ki})$,并在仿真计算完成后延迟 $S(M_{ki})$ 时间向 E_i 发送仿真结果,同时保存现场数据以备回滚处理;

④后继实体仍进行因果律检测,并在出现因果律违背时进行回滚处理.

MATW 算法一方面不要求前驱实体等待后继实体的空消息,系统并行度必定高于 NM 算法,另一方面在仿真计算完成后仍根据预测参数进行同步等待,其系统回滚律必将低于 TW 算法,性能优化的具体程度则与系统的物理模型、通信带宽以及预测估值精度等因素有关.

4.3 GDSM 算法

MATW 算法使 ATW 的基本思想适用于 SDAG,但也有如下值得改进之处: 一是它依赖于每一后继实体向前驱实体回送的延迟数据,大量回送消息的加入无疑降低了通信系统带宽; 二是如果回送数据出现滞后,必将引起前驱实体预测估值精度的下降; 三是前驱实体同步等待延迟的计算仅采用了本周期自身仿真计算延迟及后继实体过去的延迟时间序列等局部信息,若能全面及时地纳入当前周期中各个实体仿真计算与同步等待延迟的调整信息,其准确性还可进一步提高. 事实上,在 MATW 算法中只能对实体 E_i 的源始路径延迟 $D(LE_{ij})$ 的相对差进行整体预测,而 $D(LE_{ij})$ 还可进行进一步的分解:

$$D(LE_{ij}) = W(LE_{ij}) + R(LE_{ij}) + S(LE_{ij})$$

其中, $W(LE_{ij})$ 为源始路径中每个实体等待所有输入消息到达的延迟之和, $R(LE_{ij})$ 为源始路径中每个实体仿真计算时间之和, $S(LE_{ij})$ 为源始路径中每个实体同步等待延迟之和,通过对 $W(LE_{ij})$、$R(LE_{ij})$ 和 $S(LE_{ij})$ 的分项预测,便可以提高

$D(LE_{ij})$ 相对差的估计精度. 首先, 由于 SDAG 中仿真数据流构成有向无环图, 能够建立实体之间的有序关系:

命题 **5**

SDAG 中所有实体能够排列成 E_0, E_1, … E_N 的顺序, 使得若 E_k 属于 E_i 的源始路径, 则有 $k \leqslant i$.

其中 E_0 就是源, 而 E_N 便是末端, 按照这个顺序, 可以建立 SDAG 中的全局动态同步方法 GDSM:

①系统仿真启动时, 除 E_0 外各个实体的同步等待延迟均置为 0;

②仿真过程中后继实体仍进行因果律检测, 并在出现因果律违背时进行回滚处理;

③每一实体在完成仿真计算后, 均按当前的同步等待延迟向后继实体发送仿真消息;

④仿真过程中, 前驱实体发送至后继实体的仿真消息中加入本周期源始路径中各个实体的消息等待时间、仿真计算时间、和同步等待延迟数据;

⑤每一周期末端完成仿真计算后, 增加一发送至源的消息, 内容包括所有实体的消息等待时间、仿真计算时间、和同步等待延迟数据;

⑥E_0 在收到末端的实体延迟参数后, 若尚未满足预测算法初始化要求, 则在保存数据后启动下一周期仿真; 若已满足预测算法初始化要求, 则按命题 5 中顺序完成对所有实体消息等待时间、仿真计算时间的预测估计, 进而计算新的源始路径延迟相对差, 最后根据命题 4 确定等待同步延迟, 并将相应实体的同步等待延迟随同仿真消息发送各个实体.

⑦E_0 在获得自身的同步等待时间后, 完成一个周期的仿真计算及消息发送即启动下一周期的仿真而无需等待末端的回送数据.

5　测试结果与分析

5.1　主要测试条件

在 RDSP 中利用 FCT 系统结构在四个节点上进行了 NM、TW、MATW 和 GDSM 算法的仿真测试, 实验过程中关注的主要指标是单位时间内完成的仿真周期数:

$$Ct = Nu/Tt$$

其中 Tt 表示总仿真时间, Nu 表示完成的总仿真周期数, Ct 实际上反映了系统的并行度. 同时还考察了不同情形下的回滚率. 在仿真过程中, 每个实体的仿真计算时间、实体之间的通信延迟均为随机参量, 为便于仿真, 将这些随机变量的变化模型均简化为正弦函数加随机扰动的形式, 即

$$f(t) = A\sin(\omega t + \varphi) + B\varepsilon$$

其中 ε 为服从$(-1, 1)$内均匀分布的随机变量. 于是, 在$f(t)$的变化中, A 代表了稳态变化的幅度, ω 代表了稳态变化的频率, ϕ 代表了稳态变化的初始相位, B 代表了随机扰动的幅度. 采用 α- β- γ 滤波方法实现仿真计算时间和实体通信延迟的预测估计, 因而随机扰动幅度 B 与稳态变化幅度A 之比在很大程度上影响了预测算法的准确性, 为此选用了不同的比值来考察算法的性能. 还对计算时间的变化幅度与通信延迟的变化幅度、计算时间的变化周期与通信延迟的变化周期之间的不同比值进行了仿真实验.

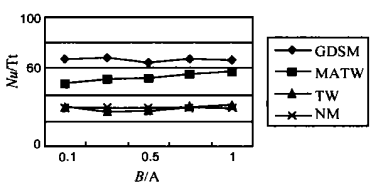

图 2　随机扰动幅度与稳态变化幅度不同比值下的系统效率

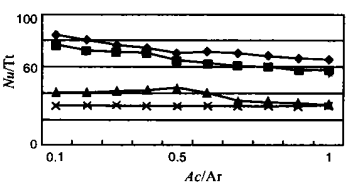

图 3　通信延迟稳态变化幅度与任务运行时间稳态变化幅度不同比值下的系统效率

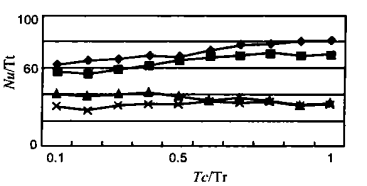

图 4　通信延迟稳态变化周期与任务运行稳态变化周期不同比值下的系统效率

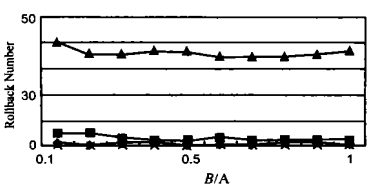

图 5　随机变化幅度与稳态变化幅度不同比值下的回流率

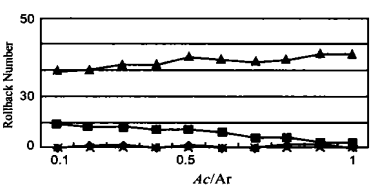

图 6　通信延迟稳态变化幅度与任务运行时间稳态变化幅度不同比值下的回流率

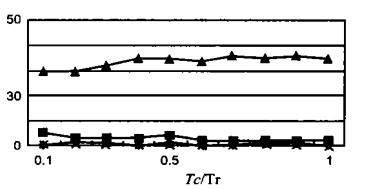

图 7　通信延迟稳态变化周期与任务运行时间稳态变化周期不同比值下的回流率

5.2　分析对比

图 2~ 7 所示测试结果显示, TW 算法与 NM 算法性能基本相当, 而 MATW 算法较之前两种算法改善约 40% 左右, GDSM 算法普遍优于 MATW 算法, 这与推断基本吻合. 尽管图中所示 GDSM 性能全面优于 MATW, 这并不说明前者完全可以替代后者, 事实上, 当末端与源之间的通信延迟预测精度大幅下降时, GDSM 性能将难以正常发挥, 而 MATW 则完全能够优于 GDSM.

6　结论

MATW 算法吸收了 ATW 算法的思想, 使其适用于 SDAG 模型, 因而同步性能较传统的 NM、TW 算法有较大程度的改善. GDSM 算法基于实体运行时间和实体间通讯时差的预测估计, 从全局角度动态调整系统中各个实体的同步延迟, 多数情况下其性能改善更优于 MATW, 这些都为具有 SDAG 模型特征的分布式仿真系统同步性能改善提供了新的途径. 然而

与ATW 算法一样,系统性能的改善在很大程度上依赖于预测参数的准确,因此根据系统通信和任务执行的变化模型寻求合适的延迟估计算法将成为MATW 和 GDSM 算法应用的关键问题.

参考文献:

[1] R M Fujimoto. Parallel discrete event simulation [J]. Communication of the ACM, 1990, 33(10): 31- 53.

[2] J B Hiller, T C Hartrum. Conservative synchronization in object-oriented parallel battlefield discrete event simulation [A]. Proc. 11th Workshop on Parallel and Distributed Simulation [C]. 1997. 12- 19.

[3] T K Som, R G Sargent. A probabilistic event scheduling policy for optimistic parallel discrete event simulation [A]. Proc. Twelfth Workshop on Parallel and Distributed Simulation [C]. 1998. 56- 63.

[4] K M Chandy, J Misra. Distributed simulation: A case study in design and verification of distributed programs [J]. IEEE Trans. on Software Engineering, 1979, 5(5): 440- 452.

[5] D R Jefferson, H Sowizral. Fast concurrent simulation using the time warp mechanism, Part I: local control [R]. Technical Report N-1906-AF, RAND Corp, 1982.

[6] A Ferscha. Adaptive time warp simulation of timed petri nets [J]. IEEE Trans. on Software Engineering, 1999, 25(2): 237- 257.

[7] 许建峰, 朱晴波, 胡宁, 谢立. 分布式实时系统中的预测调度算法 [J]. 软件学报, 2000, 11(1): 95- 103.

作者简介:

许建峰 男, 1961 年生于江苏如皋, 博士研究生, 主要研究领域为雷达软件工程和分布式系统.

朱晴波 男, 1976 年生于江苏无锡, 硕士研究生, 主要研究方向为分布式操作系统.